

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
8 February 2001 (08.02.2001)

PCT

(10) International Publication Number
WO 01/09740 A1

(51) International Patent Classification⁷: G06F 15/173

(21) International Application Number: PCT/US00/21037

(22) International Filing Date: 2 August 2000 (02.08.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/366,657 3 August 1999 (03.08.1999) US

(71) Applicant: SONY ELECTRONICS INC. [US/US]: 1
Sony Drive, Park Ridge, NJ 07656 (US).

(74) Agents: KOERNER, Gregory, J. et al.; Simon & Koerner LLP, Suite B, 10052 Pasadena Avenue, Cupertino, CA 95014 (US).

(81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

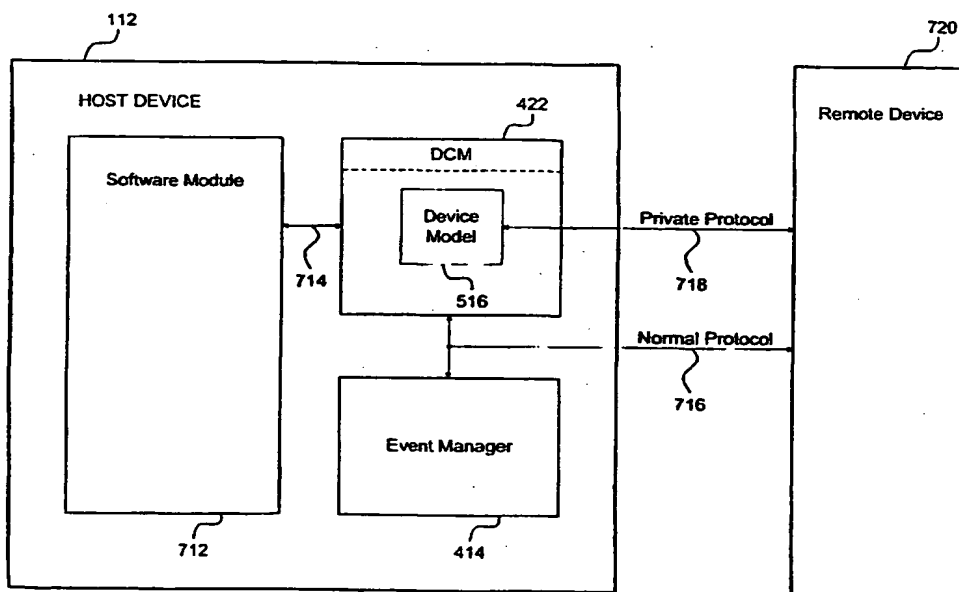
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— With international search report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM AND METHOD FOR IMPLEMENTING DEVICE MODELS IN AN ELECTRONIC NETWORK



(57) Abstract: A system and method for implementing device models (516) in an electronic network (110) comprises a device control module (422) that maintains a local device model (516) to accurately represent various device states (614) of a remote device (720) on the electronic network (110). The device control module (422) may then efficiently analyze the local device model (516) to provide updated information about the remote device (720) to local software module (714).

WO 01/09740 A1

**SYSTEM AND METHOD FOR IMPLEMENTING DEVICE MODELS
IN AN ELECTRONIC NETWORK**

CROSS-REFERENCE TO RELATED APPLICATIONS

5

This application relates to co-pending U.S. Patent Application No. 09/257,344, entitled "System And Method For Implementing Active Registries In An Electronic Network," filed on February 25, 1999, to co-pending U.S. Patent Application No. _____, entitled "System And
10 Method For Discovering Extended Capabilities Of Devices In An Electronic Network," filed on _____, and to co-pending U.S. Patent Application No. 09/289,500, entitled "System And Method For Maintaining Fully-Replicated Registries In An Electronic Network," filed on April 9, 1999, which are hereby incorporated by reference. The foregoing cross-referenced
15 applications are commonly assigned.

BACKGROUND OF THE INVENTION

20

1. Field of the Invention

This invention relates generally to electronic networks, and relates more particularly to a system and method for implementing device models in an electronic network.

25

2. Description of the Background Art

30

Implementing an effective method for managing communications between electronic devices within an electronic network is a significant consideration for manufacturers and designers of contemporary electronic
30 systems. An electronic device in a distributed electronic network may advantageously communicate with other remote electronic devices in the network to share and substantially increase the resources available to

individual devices in the network. For example, an electronic network may be implemented in a user's home to enable flexible and beneficial sharing of resources between various consumer electronic devices, such as personal computers, digital video disk devices, digital set-top boxes for digital broadcasting, television sets, and audio playback systems.

Managing communications in a network of electronic devices may create substantial challenges for designers of electronic networks. For example, enhanced demands for increased functionality and performance may require more system processing power and require additional hardware resources across the network. An increase in processing or hardware requirements may also result in a corresponding detrimental economic impact due to increased production costs and operational inefficiencies.

Network size is also a factor that affects the management of communications in an electronic network. Communications in an electronic network typically become more complex as the number of individual devices or nodes increases. Assume that a particular device on an electronic network is defined as a local device with local software elements, and other devices on the electronic network are defined as remote devices with remote software elements. Accordingly, a local software module on the local device may need to communicate with various remote software elements on remote devices across the electronic network. However, successfully managing a substantial number of electronic devices across a single network may provide significant benefits to a system user.

Furthermore, enhanced device capability to perform various advanced functions may provide additional benefits to a system user, but may also place increased demands on the control and management of the various devices in the electronic network. For example, an enhanced electronic network that effectively accesses, processes, and displays digital television programming may benefit from efficient network communication techniques because of the large amount and complexity of the digital data involved.

Therefore, for all the foregoing reasons, implementing an efficient method for managing communications between electronic devices in a

distributed electronic network remains a significant consideration for designers, manufacturers, and users of electronic systems.

SUMMARY OF THE INVENTION

In accordance with the present invention, a system and method are disclosed for implementing device models in an electronic network. In one embodiment of the invention, network software in a local host device may utilize any appropriate techniques to maintain a device model corresponding to a remote device on the electronic network. For example, a local software module in a host device may generate a device request to a device control module (DCM). In response, the DCM preferably translates the device request into an underlying device command that a remote device on the electronic network may then receive using a normal protocol.

The remote device then preferably returns a command acknowledge event to a local event manager using the foregoing normal protocol. The command acknowledge event preferably notifies the event manager that the device command initially sent from the DCM has been received and executed by the remote device. The event manager preferably includes a series of notification registrations from various software elements in the host device to request a notification message upon the occurrence of specified corresponding events. In accordance with the present invention, the DCM initially registers with the event manager for a notification message whenever a command acknowledge event is transmitted by the remote device.

In the present embodiment, a model manager for a device model which corresponds to the remote device may therefore access the notification message from the event manager, and responsively update a corresponding device state in the device model. To ensure correct operation of the device model, the remote device preferably provides additional self-initiated notification messages to the DCM for any device state changes that are not caused by device commands from the DCM.

In accordance with the present invention, the foregoing techniques may be augmented through the use of an efficient polling process during which the device model preferably formulates and propagates a polling

inquiry to the remote device using a private protocol. The remote device preferably returns a polling reply to the device model in response to the polling inquiry. The DCM may then advantageously utilize the polling reply to update a corresponding device state in the device model to thereby
5 compensate for any drift or variation between the current status of the remote device and corresponding device states of the device model.

The remote device may also advantageously provide significant-event notifications to the DCM whenever a defined significant event occurs. The DCM may thus utilize the foregoing significant-event notifications to update
10 the device model and thereby reduce the amount of polling required to maintain the device model.

In accordance with the present invention, the device model may then be advantageously utilized by any module, element, or device to provide local information about the current status of the remote device. For
15 example, the DCM may effectively utilize the device model to efficiently respond to various types of queries about the current status of the remote device without actually communicating directly with the remote device for every individual query.

The DCM may thus return a rapid query reply to a querying software
20 module without unduly burdening the electronic network with unnecessary messaging traffic. Furthermore, the foregoing local query technique efficiently conserves valuable device resources of the remote device in order to perform other processing tasks.

The foregoing embodiment is discussed in the context of a single
25 device model, however the present invention may readily be utilized to implement any number of device models across the electronic network. The present invention thus provides a system and method to efficiently implement device models across an electronic network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram for one embodiment of an electronic network, in accordance with the present invention;

5

FIG. 2 is a block diagram for one embodiment of an exemplary device from FIG. 1, in accordance with the present invention;

FIG. 3 is a diagram for one embodiment of the memory of FIG. 2, in accordance with the present invention;

10

FIG. 4 is a diagram for one embodiment of the network software of FIG. 3, in accordance with the present invention;

FIG. 5 is a diagram for one embodiment of a device control module of FIG. 4, in accordance with the present invention;

15

FIG. 6 is a diagram for one embodiment of the device model of FIG. 6, in accordance with the present invention;

20

FIG. 7 is a block diagram illustrating the use of a device model, in accordance with one embodiment of the present invention;

FIG. 8 is a flowchart of method steps for performing a query process, in accordance with one embodiment of the present invention;

25

FIG. 9 is a flowchart of method steps for maintaining a device model, in accordance with one embodiment of the present invention; and

FIG. 10 is a flowchart of method steps for using a private protocol to poll a remote device, in accordance with one embodiment of the present invention.

30

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention relates to an improvement in electronic network technology. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. Thus, the present invention is not intended to be limited to the embodiment shown, but is to be accorded the widest scope consistent with the principles and features described herein.

The present invention comprises a system and method for efficiently implementing device models in an electronic network, and preferably includes a device control module that maintains a local device model to accurately represent various device states of a remote device on the electronic network. The device control module may then efficiently analyze the local device model to provide updated information about the remote device to local software modules.

Referring now to FIG. 1, a block diagram for one embodiment of an electronic network 110 is shown, in accordance with the present invention. In the FIG. 1 embodiment, network 110 includes, but is not limited to, device A 112(a), device B 112(b), device C 112(c), and device D 112(d). In other embodiments, network 110 may readily be implemented using a larger or smaller number of devices than the four devices (device A 112(a) through device D 112(d)) shown in the FIG. 1 embodiment.

In the FIG. 1 network 110, device A 112(a), device B 112(b), device C 112(c), and device D 112(d) preferably communicate with each other through a commonly-shared network bus 120. In the FIG. 1 embodiment, network bus 120 is preferably implemented according to the IEEE 1394

interconnectivity standard. However, in alternate embodiments, other appropriate and compatible interconnectivity standards are also contemplated for use in conjunction with the present invention.

In the FIG. 1 embodiment, network 110 may preferably be configured to operate in accordance with the Home Audio/Video Interoperability (HAVi) core specification (version 1.0 beta, at www.HAVi.org) which is hereby incorporated by reference. Therefore, device A 112(a), device B 112(b), device C 112(c), and device D 112(d) may be implemented as various types of consumer electronics devices, including, but not limited to, personal computers, digital video disk devices, television sets, audio reproduction systems, video tape recorders (VCRs), and set-top boxes for digital video broadcasting. However, in various alternate embodiments, network 110 may readily be implemented as any appropriate electronic network configured to permit communication between any desired types of electronic devices.

In the FIG. 1 embodiment, the various electronic devices that form part of network 110 preferably include the following four categories of electronic devices: full devices (FD), intermediate devices (ID), base devices (BD), and legacy devices (LD). The foregoing four categories of electronic devices (FD, ID, BD, and LD) are further discussed below in conjunction with FIGS. 2 and 3. In alternate embodiments of the present invention, network 110 may readily include various other categories of electronic devices in addition to, or instead of, the four categories of FD, ID, BD, and LD.

25

Referring now to FIG. 2, a block diagram for one embodiment of an exemplary device 112 from FIG. 1 is shown, in accordance with the present invention. In the FIG. 2 embodiment, device 112 preferably includes, but is not limited to, a processor 212, an input/output interface (I/O) 214, and a memory 216 that are each coupled to, and communicate with each other via, a common device bus 218. In the FIG. 2 embodiment, device 112 is

30

preferably configured to represent either a full device or an intermediate device, as referred to above in the discussion of the FIG. 1 network 110.

In the FIG. 2 embodiment, processor 212 may be implemented to include any appropriate and compatible generic, multi-purpose
5 microprocessor device. The FIG. 2 input/output interface (I/O) 214 preferably provides an effective interface to facilitate communications between device 112 and network bus 120 (FIG. 1). In the FIG. 2 embodiment, memory 216 may be implemented to include any combination of desired storage devices, including, but not limited to, read-only memory
10 (ROM), random-access memory (RAM), and various types of non-volatile memory, such as floppy disks or hard disks. The contents and functionality of memory 216 are further discussed below in conjunction with FIGS. 3 and 4.

15 Referring now to FIG. 3, a diagram for one embodiment of FIG. 2 memory 216 is shown, in accordance with the present invention. In the FIG. 3 embodiment, memory 216 includes one or more device applications 312, a network application program interface (API) 314, network software 316, self-describing data (SDD) 320, a device driver 318, a platform-specific
20 application program interface (API) 322, and a vendor-specific platform 324. In alternate embodiments, memory 216 may readily include various components and elements that are different from, or in addition to, those software components 312 through 324 discussed in conjunction with the FIG. 3 embodiment.

25 In the FIG. 3 embodiment, device application(s) 312 preferably include software instructions that are executed by processor 212 (FIG. 2) to effectively manage and control the functionality of device 112. Network API 314 preferably serves as an interface between various elements of network software 316 and device application 312.

30 In the FIG. 3 embodiment, network software 316 preferably includes one or more software elements that are executed by processor 212 to advantageously permit device 112 to communicate and cooperate with other

devices in network 110. The contents and functionality of network software 316 are further discussed below in conjunction with FIG. 4.

Self-describing data (SDD) 320 preferably includes various types of relevant information regarding device 112. For example, SDD 320 may include information specifying the manufacturer, model, version, serial number, and other fixed data that specifically corresponds to device 112. Device driver 318 preferably includes appropriate software instructions that permit device 112 to communicate with network bus 120 (FIG. 1).

In the FIG. 3 embodiment, platform-specific API 322 provides an interface that preferably permits network software 316 to communicate with vendor-specific platform 324. In the FIG. 3 embodiment, vendor-specific platform 324 may include basic operating system software for supporting low-level operations of device 112.

The FIG. 3 embodiment of memory 216 typically corresponds to a full device (or FD, as discussed above in conjunction with FIG. 1) that preferably includes a complete set of network software 316 to permit optimal compatibility and functionality with network 110. Alternately, memory 216 may correspond to an intermediate device (ID) which includes only a reduced set of software elements from network software 316. In contrast, a base device (BD) is preferably hosted on network 110 by a full device or an intermediate device, and therefore typically does not include network software 316. A base device, however, preferably does include self-describing data 320 and a device driver 318.

A legacy device (LD) may be defined as a device that does not comply with the architectural specifications of network 110 and network software 316. Legacy devices typically were designed and manufactured prior to the design and implementation of network 110 and network software 316. Therefore, a legacy device is preferably hosted on network 110 by a full device or an intermediate device, and typically does not include network software 316 or self-describing data 320. A digital base device, however, may include a device driver 318 for interfacing with network bus 120. Full devices, intermediate devices, base devices, and legacy devices are further

discussed in the Home Audio/Video Interoperability (HAVi) core specification (see pages 3 through 22) that has been previously incorporated by reference.

5 Referring now to FIG. 4, a diagram for one embodiment of the network software 316 of FIG. 3 is shown, in accordance with the present invention. In the FIG. 4 embodiment, network software 316 preferably comprises a number of software elements, including a registry 412, an event manager 414, a device control module (DCM) manager 416, a stream manager 418, a
10 resource manager 420, one or more device control modules (DCMs) 422 and one or more corresponding functional control modules (FCMs) 423, a messaging system 424, and a communication media manager (CMM) 426.

In the FIG. 4 embodiment, software elements 412 through 426 are preferably configured to function in accordance with the Home Audio/Video
15 Interoperability (HAVi) architecture which has previously been incorporated herein by reference. However, in alternate embodiments, network software 316 may readily conform to any other appropriate and compatible interoperability architecture, and may also include various software elements that are different from, or in addition to, those elements 412
20 through 426 that are presented in the FIG. 4 embodiment.

In the FIG. 4 embodiment, registry 412 may preferably include a listing of software elements in network software 316. Registry 412 also preferably may include relevant element information or attributes corresponding to the listed software elements. For example, elements 412
25 through 426 from network software 316 and corresponding element information may be listed in registry 412. Registry 412 therefore may serve as a directory service for applications 312 or software elements in network 110. Registry 412 may thus allow any application or software element to obtain a software element identifier (SEID) for identifying and locating
30 another software element in network 110. In accordance with the present invention, registry 412 may also include a remote registry list that identifies all remote registries on network 110.

In the FIG. 4 embodiment, event manager 414 preferably serves as a network-event notification service to notify various software elements (that have previously subscribed for notification) about the occurrence of a specified network event, such as a change in a software element or a change in network 110. DCM manager 416 is preferably responsible for installing and removing DCMs 422 on full devices or intermediate devices. Stream manager 418 is preferably responsible for managing real-time transfer of data and other information between various functional components of network 110.

In the FIG. 4 embodiment, resource manager 420 preferably facilitates the sharing of various resources and scheduling of various actions in network 110. A device control module (DCM) 422 preferably includes a software element that is used to control a specific associated device on network 110. A given DCM 422 preferably includes one or more directly-corresponding functional control modules (FCMs) 423 that each control a specific functional component within the particular device 112 that corresponds to the FCM 423. A full device or an intermediate device may preferably host a DCM 422 to control a remote base device or a remote legacy device on network 110. In the FIG. 4 embodiment, messaging system 424 is preferably responsible for bi-directionally transferring various messages between the software elements of network software 316. Communication media manager (CMM) 426 coordinates and manages asynchronous and isochronous communications through device driver 318 onto network bus 120.

Network software 316 preferably performs a number of significant and related operations whenever a particular device is removed from, or added to, network 110. When a device is added or removed from network 110, then network bus 120 preferably triggers a bus reset event which notifies all connected devices about the change in network 110. Following the bus reset event, all DCM managers 416 in network 110 preferably perform a negotiation procedure to determine which, if any, DCM manager 416 is the most appropriate host for controlling the newly-added device 112. Each

DCM manager 416 in network 110 may therefore maintain a current list of all devices in network 110. Once a given DCM manager 416 is selected as host, that host DCM manager 416 responsively instantiates a new DCM 422 as an abstraction of the control interface of the newly-added device.

- 5 Network software 316 preferably also updates relevant software element information in registry 412 whenever a device is removed from, or added to, network 110.

Referring now to FIG. 5, a diagram for one embodiment of FIG. 4
10 device control module 422 is shown, in accordance with the present invention. In the FIG. 5 embodiment, DCM 422 includes, but is not limited to, a request manager 512, a query manager 514, and a device model 516. In alternate embodiments, DCM 422 may include various elements that are different from, or in addition to, those discussed in conjunction with the
15 FIG. 5 embodiment.

In the FIG. 5 embodiment, DCM manager 416 preferably instantiates
DCM 422 in a local host device 112 to represent and control another hosted
remote device on network 110. The remote device is typically a base device
or a legacy device, as discussed above in conjunction with FIG. 3. DCM 422
20 preferably comprises an abstraction of the control interface of the remote device that may then be utilized to interact between a local software module and the remote device.

In the FIG. 5 embodiment, a local software module (such as device
application(s) 312 or any other software element of network software 316)
25 may utilize request manager 512 to control a corresponding remote device of network 110. In practice, the software module preferably transmits a device request to DCM 422. For example, if the remote device is a VCR, then the software module may transmit a VCR "play" request to DCM 422. In response, DCM 422 translates the device request into an underlying
30 device command, and then propagates the device command to the remote device for appropriate action.

Similarly, in the FIG. 5 embodiment, a local software module (such as device application(s) 312 or any other software element in network software 316) may utilize query manager 514 to perform a query process regarding a corresponding remote device of network 110. In practice, the software module preferably transmits a device query to DCM 422. In response, DCM 422 preferably parses the device query, and then analyzes device model 516 to return an appropriate query response to the querying software module, in accordance with the present invention.

In the FIG. 5 embodiment, device model 516 includes a local representation of various operational states and parameters of the corresponding remote device of network 110. In alternate embodiments, device model 516 may readily be implemented as part of another software element (such as FCM 423) on network 110, or as an independent module on network 110. Furthermore, in certain embodiments, device model 516 may represent various devices on network 110 other than the foregoing hosted remote device. The maintenance and utilization of device model 516 is further discussed below in conjunction with FIGS. 6 through 10.

Referring now to FIG. 6, a diagram for one embodiment of the FIG. 5 device model 516 is shown, in accordance with the present invention. In the FIG. 6 embodiment, device model 516 includes, but is not limited to, a state data structure 612, a series of state simulation routines 616 (state simulation routine 1 (616(a)) through state simulation routine N (616(c)), and a model manager 618. In alternate embodiments, device model 516 may be configured to include various elements that are different from, or in addition to, those discussed in conjunction with the FIG. 6 embodiment.

In the FIG. 6 embodiment, state data structure 612 is preferably implemented as a detailed local model of a corresponding remote device on network 110. State data structure 612 preferably includes a series of separate device states 614 (device state 1 (614(a)) through device state N (614(c))). Each one of the device states 614 preferably corresponds to a separate functional state, parameter, attribute, or any other characteristic

of the corresponding remote device. For example, a given device state 614 may correspond to a tape counter value in a remote VCR device. In alternate embodiments, state data structure 612 may be configured in any other appropriate manner. For example, state data structure 612 may be
5 configured to include a detailed device state machine or various other types of detailed device representations.

In the FIG. 6 embodiment, each one of the state simulation routines 616 preferably corresponds to a separate one of the device states 614. However, certain of the device states 614 may not relate to any of the state
10 simulation routines 616. In accordance with the present invention, each of the state simulation routines 616 preferably comprises a series of program instructions that mimic relevant expected performance characteristics of a corresponding remote device, to thereby generate simulation update values for a given device state 614. For example, a state simulation routine 616
15 may mimic the expected operation of a tape counter in a VCR device, and then responsively generate simulated counter update values for a corresponding device state 614.

In the FIG. 6 embodiment, model manager 618 preferably includes a series of program instructions for controlling and managing the operation of
20 device model 516. For example, model manager 618 may periodically perform update procedures on selected device states 614 in state data structure 612 to thereby maintain device model 516 in a current and accurate condition. In alternate embodiments, model manager 618 may readily be implemented as a discrete program module that is separate from
25 device model 516. The maintenance and utilization of device model 516 is further discussed below in conjunction with FIGS. 7 through 10.

Referring now to FIG. 7, a block diagram illustrating use of the FIG. 6 device model 516 is shown, in accordance with one embodiment of the
30 present invention. In the FIG. 7 embodiment, a local software module 712 (for example, device application 312 or any other software element of network software 316) may generate a device request to device control

module (DCM) 422 via path 714. In response, DCM 422 preferably translates the device request into an underlying device command that a remote device 720 on network 110 may then receive using a normal protocol via path 716.

5 Remote device 720 then preferably returns a command acknowledge event to event manager 414 using the normal protocol via path 716. The command acknowledge event preferably notifies event manager 414 that the device command initially sent from DCM 422 has been received and executed by remote device 720. Event manager 414 preferably includes a
10 series of notification registrations from various software elements in host device 112 to request a notification message upon the occurrence of specified corresponding events. In accordance with the present invention, DCM 422 initially registers with event manager 414 for a notification message whenever a command acknowledge event is transmitted by remote
15 device 720. The foregoing notification message preferably includes information specifying characteristics of the device command that was received and executed by remote device 720.

Model manager 618 may therefore access the notification message from event manager 414 and responsively update a corresponding device
20 state 614 in state data structure 612 to effectively maintain device model 516. In an alternate embodiment, model manager 618 may similarly directly utilize the foregoing device request from software module 712 to update device model 516, instead of using the notification message from event manager 414. To ensure correct operation, remote device 720
25 preferably provides additional self-initiated notification messages to DCM 422 for any device state changes that are not caused by device commands from DCM 422.

In the FIG. 7 embodiment, the foregoing techniques may be augmented through the use of an efficient polling process during which
30 device model 516 preferably formulates and propagates a polling inquiry to remote device 720 using a private protocol via path 718. For example, if remote device is a VCR device, then device model 516 may periodically poll

remote device 720 to determine the actual current status of a VCR tape counter. In the FIG. 7 embodiment, for reasons of illustration, the private protocol and the normal protocol are shown on separate paths (paths 718 and 716, respectively). However, in practice, the private protocol and the
5 normal protocol may alternately be transmitted over the same path (for example, by utilizing network bus 120).

Remote device 720 preferably returns a polling reply to device model 516 in response to the polling inquiry. For example, the foregoing remote VCR device would preferably return a polling reply that specifies the current
10 status of the VCR tape counter. DCM 422 may then advantageously utilize the polling reply to update the corresponding device state 614 to thereby compensate for any drift or variation between the current state of remote device 720 and corresponding device states 614 of device model 516. The private protocol of the foregoing polling process may utilize any appropriate
15 messaging techniques and preferably depends upon various specific mechanisms and characteristics associated with remote device 720.

In the FIG. 7 embodiment, remote device 720 may also advantageously provide significant-event notifications to DCM 422 whenever a defined significant event occurs. DCM 422 may thus utilize the foregoing
20 significant-event notifications to update device model 516 and thereby reduce the amount of polling required to maintain device model 516. In alternate embodiments, device model 516 may be implemented as a simple abstraction of remote device 720, rather than a detailed model. In such cases, the foregoing polling process may be similarly utilized to effectively
25 maintain the simplified version of device model 516.

In accordance with the present invention, device model 516 of DCM 422 may therefore be advantageously utilized by any appropriate module, element, or device to provide local information about the current status of remote device 720. For example, DCM 422 may effectively utilize device
30 model 516 to efficiently respond to various types of queries about the current status of remote device 720 without actually communicating directly with remote device 720 for every individual query. In certain

applications, a software module 712 may wish to query remote device 720 at very frequent intervals, and therefore, local query analysis of device model 516 may provide significantly increased efficiency across network 110.

5 In the FIG. 7 embodiment, a software module 712 may propagate a device query to DCM 422 via path 714. In the absence of device model 516, DCM 422 would then be required to perform the burdensome and time-consuming process of propagating the query to remote device 720 via path 716 (possibly at frequent intervals). Remote device 720 would also be
10 required to utilize valuable device resources to perform the query and return a query reply to DCM 422 via path 716.

However, in accordance with the present invention, DCM 422 may locally analyze device model 516 to efficiently perform the query from software module 712 without communicating with remote device 720. DCM
15 422 may thus return a rapid query reply to software module 712 without unduly burdening network 110 with unnecessary messaging traffic. Furthermore, the foregoing local query technique efficiently conserves valuable device resources of remote device 720 in order to performing other processing tasks.

20

Referring now to FIG. 8, a flowchart of method steps for performing a query process is shown, in accordance with one embodiment of the present invention. In the FIG. 8 embodiment, initially, in step 812, DCM manager 416 instantiates a device control module (DCM) 422 corresponding to a
25 remote device 720 on network 110. In the FIG. 8 embodiment, DCM 422 preferably includes a device model 516 corresponding to remote device 720.

Then, in step 814, DCM 422 maintains device model 814 to accurately represent remote device 720, as discussed in conjunction with FIG. 7 and FIG. 9. Next, in step 816, DCM 422 determines whether a query
30 concerning the status of remote device 720 has been sent from a software module 712 to DCM 422. If no query has been sent, then DCM 422 continues to monitor for a query regarding remote device 720.

However, in step 816, if a query concerning the status of remote device 720 has been sent from a software module 712 to DCM 422, then, in step 818, DCM 422 advantageously performs the query by analyzing local device model 516. Finally, in step 820, DCM 422 returns a query reply to the software module 712 to efficiently complete the FIG. 8 query process.

Referring now to FIG. 9, a flowchart of method steps for maintaining a device model 516 is shown, in accordance with one embodiment of the present invention. In the FIG. 9 embodiment, initially, in step 912, model manager 618 of device model 516 preferably determines and stores initial device states 614 into state data structure 612 of device model 516.

In step 914, processor 212 of host device 112 begins executing the state simulation routines 616 of device model 516 to simulate various device states 614 of remote device 720. Then, in step 916, model manager 618 determines whether any new device states 614 have been received by DCM 422. The foregoing new device states 614 may originate from any desired source, and may be generated using any appropriate technique. For example, DCM 422 may update devices state 614 in device model 516 using simulated update values from the state simulation routines 616 of foregoing step 914. DCM 422 may also update devices state 614 by using polling reply values generated through a polling process symbolized by the reference letter "A" of FIG. 9. The foregoing polling process is further discussed below in conjunction with FIG. 10. In addition, DCM 422 may update devices state 614 using any of the various techniques discussed above in conjunction with FIG. 7.

Therefore, in step 916, model manager 618 preferably determines whether new device states 614 have been received from any source. If no new device states 614 have been received, then model manager 618 continues to monitor for new device states 614. However, in step 916, if any new device states 614 have been received, then, in step 918, model manager 618 advantageously updates state data structure 612 of device model 516 with the appropriate new device states 614. The FIG. 9 process

then returns to step 916, where model manager 618 continues to monitor for new device states 614 from any source.

Referring now to FIG. 10, a flowchart of method steps for using a private protocol to poll a remote device 720 is shown, in accordance with one embodiment of the present invention. In the FIG. 10 embodiment, initially, in step 1012, DCM 422 defines a polling threshold. For example, the foregoing polling threshold may be a finite time period that is optimized depending upon a particular corresponding device state 614.

In step 1014, DCM 422 determines whether the defined polling threshold has been reached. Next, in step 1016, if the polling threshold of step 1012 has been reached, then, in step 1016, DCM 422 preferably sends a polling inquiry to remote device 720 using a private protocol, as discussed above in conjunction with FIG. 7. Finally, in step 1018, DCM 422 advantageously receives a polling reply from remote device 720 using the foregoing private protocol. The FIG. 10 method then proceeds to reference letter "A" which is located immediately prior to step 916 of FIG. 9.

The invention has been explained above with reference to a preferred embodiment. Other embodiments will be apparent to those skilled in the art in light of this disclosure. For example, the present invention may readily be implemented using configurations and techniques other than those described in the preferred embodiment above. Additionally, the present invention may effectively be used in conjunction with systems other than the one described above as the preferred embodiment. Therefore, these and other variations upon the preferred embodiments are intended to be covered by the present invention, which is limited only by the appended claims.

WHAT IS CLAIMED IS:

1. A system for efficiently implementing an electronic network (110),
comprising:
 - 5 a device model (516) coupled to said electronic network (110), said
device model (516) representing a remote device (720); and
a software module (714) configured to reference said device model
(516) to thereby obtain device information corresponding to said
remote device (720).
- 10 2. The system of claim 1 wherein said software module (714) is a device
application (312) that communicates with said remote device (720) on said
electronic network (110).
- 15 3. The system of claim 1 wherein said device model (516) forms part of
network software (316) for said electronic network (110).
4. The system of claim 3 wherein said network software (316) is
configured to comply with a home audio-video interoperability specification.
- 20 5. The system of claim 1 wherein said electronic network (110) is
connected through a network bus (120) configured using an IEEE 1394
interconnectivity standard.
- 25 6. The system of claim 1 wherein said software module (714) and said
device module form part of a local set-top box device on said electronic
network (110).
7. The system of claim 1 wherein said device information includes
30 individual device states (614) corresponding to said remote device (720) on
said electronic network (110).

8. The system of claim 1 wherein a device control module (422) maintains said device model (516) to accurately represent a current status of said remote device (720), and wherein said device control module (422) responds to a query from said software module (714) by locally analyzing
5 said device model (516) to thereby provide a rapid query response while simultaneously minimizing messaging traffic on said electronic network (110) and conserving processing resources in said remote device (720).
9. The system of claim 1 wherein a manager module (416) instantiates a
10 device control module (422) in a host device (112), said device control module (422) including said device model (516).
10. The system of claim 9 wherein said device model (516) comprises a state data structure (612), state simulation routines (616), and a model
15 manager (618), said state data structure (612) including device states (614) corresponding to said remote device (720).
11. The system of claim 10 wherein said model manager (618) determines and stores initial values for said device states (614) into said state data
20 structure (612).
12. The system of claim 10 wherein a processor (212) in said host device (112) executes said state simulation routines (616) to generate simulation values for updating said device states (614), said state simulation routines
25 (616) simulating expected performance characteristics of said remote device (720).
13. The system of claim 10 wherein said device control module (422) translates device requests into underlying device commands and provides
30 said device commands to said remote device (720) for execution.

14. The system of claim 13 wherein said remote device (720) returns a command acknowledgment event to an event manager (414) that responsively propagates an event notification to said device control module (422) based on a notification registration from said device control module (422).

15. The system of claim 14 wherein said model manager (618) updates said state data structure (612) based upon said event notification from said event manager (414).

16. The system of claim 10 wherein said device control module (422) sends a polling inquiry to said remote device (720) using a private protocol (718) whenever a polling threshold is reached.

17. The system of claim 16 wherein said remote device (720) returns a polling reply to said device control module (422) using said private protocol (718), said model manager (618) responsively updating said device model (516) using said polling reply.

18. The system of claim 9 wherein said remote device (720) transmits a self-initiated device-state update message to said device control module (422) to report selected significant device-state changes, or to report ancillary device-state changes that are not initiated through said device control module (422).

19. The system of claim 13 wherein said model manager (618) updates said state data structure (612) of said device model (516) based on new device states (614) received from any source, including said device requests received directly from said software module (714).

20. The system of claim 10 wherein said device control module (422) locally responds to a query from said software module (714) about said remote device (720) by performing a corresponding query analysis on said state data structure (612), and then generating a local query reply to said software module (714).

21. A method for efficiently implementing an electronic network (110), comprising the steps of:

maintaining a device model (516) to represent a remote device (720)

on said electronic network (110); and

analyzing said device model (516) to thereby provide device information to a software module (714), said device information corresponding to said remote device (720).

22. The method of claim 21 wherein said software module (714) is a device application (312) that communicates with said remote device (720) on said electronic network (110).

23. The method of claim 21 wherein said device model (516) forms part of network software (316) for said electronic network (110).

24. The method of claim 23 wherein said network software (316) is configured to comply with a home audio-video interoperability specification.

25. The method of claim 21 wherein said electronic network (110) is connected through a network bus (120) configured using an IEEE 1394 interconnectivity standard.

26. The method of claim 21 wherein said software module (714) and said device module form part of a local set-top box device on said electronic network (110).

27. The method of claim 21 wherein said device information includes individual device states (614) corresponding to said remote device (720) on said electronic network (110).

5 28. The method of claim 21 wherein a device control module (422) maintains said device model (516) to accurately represent a current status of said remote device (720), and wherein said device control module (422) responds to a query from said software module (714) by locally analyzing said device model (516) to thereby provide a rapid query response while
10 simultaneously minimizing messaging traffic on said electronic network (110) and conserving processing resources in said remote device (720).

29. The method of claim 21 wherein a manager module (416) instantiates a device control module (422) in a host device (112), said device control
15 module (422) including said device model (516).

30. The method of claim 29 wherein said device model (516) comprises a state data structure (612), state simulation routines (616), and a model manager (618), said state data structure (612) including device states (614)
20 corresponding to said remote device (720).

31. The method of claim 30 wherein said model manager (618) determines and stores initial values for said device states (614) into said state data structure (612).

25 32. The method of claim 30 wherein a processor (212) in said host device (112) executes said state simulation routines (616) to generate simulation values for updating said device states (614), said state simulation routines (616) simulating expected performance characteristics of said remote device
30 (720).

33. The method of claim 30 wherein said device control module (422) translates device requests into underlying device commands and provides said device commands to said remote device (720) for execution.

5 34. The method of claim 33 wherein said remote device (720) returns a command acknowledgment event to an event manager (414) that responsively propagates an event notification to said device control module (422) based on a notification registration from said device control module (422).

10

35. The method of claim 34 wherein said model manager (618) updates said state data structure (612) based upon said event notification from said event manager (414).

15 36. The method of claim 30 wherein said device control module (422) sends a polling inquiry to said remote device (720) using a private protocol (718) whenever a polling threshold is reached.

37. The method of claim 36 wherein said remote device (720) returns a
20 polling reply to said device control module (422) using said private protocol (718). said model manager (618) responsively updating said device model (516) using said polling reply.

38. The method of claim 29 wherein said remote device (720) transmits a
25 self-initiated device-state update message to said device control module (422) to report selected significant device-state changes, or to report ancillary device-state changes that are not initiated through said device control module (422).

30

39. The method of claim 33 wherein said model manager (618) updates said state data structure (612) of said device model (516) based on new device states (614) received from any source, including device requests received directly from said software module (714).

5

40. The method of claim 30 wherein said device control module (422) locally responds to a query from said software module (714) about said remote device (720) by performing a corresponding query analysis on said state data structure (612), and then generating a local query reply to said software module (714).

10

41. The method of claim 37 wherein said private protocol (718) is based upon mechanisms and characteristics provided by said remote device (720).

15 42. The method of claim 30 wherein said device model (516) and said model manager (618) do not form part of said device control module (422).

43. The method of claim 21 wherein said device model (516) includes a detailed state machine representation of said remote device (720).

20

44. The method of claim 21 wherein said device model (516) includes a simplified abstraction of said remote device (720).

25 45. A system for efficiently implementing an electronic network (110), comprising:

means for maintaining a device model (516) to represent a remote device (720) on said electronic network (110); and

means for analyzing said device model (516) to thereby provide device information to a software module (714), said device information corresponding to said remote device (720).

30

46. A computer-readable medium comprising program instructions for efficiently implementing an electronic network (110) by performing the steps of:

- 5 maintaining a device model (516) to represent a remote device (720)
 on said electronic network (110); and
 analyzing said device model (516) to thereby provide device
 information to a software module (714), said device information
 corresponding to said remote device (720).

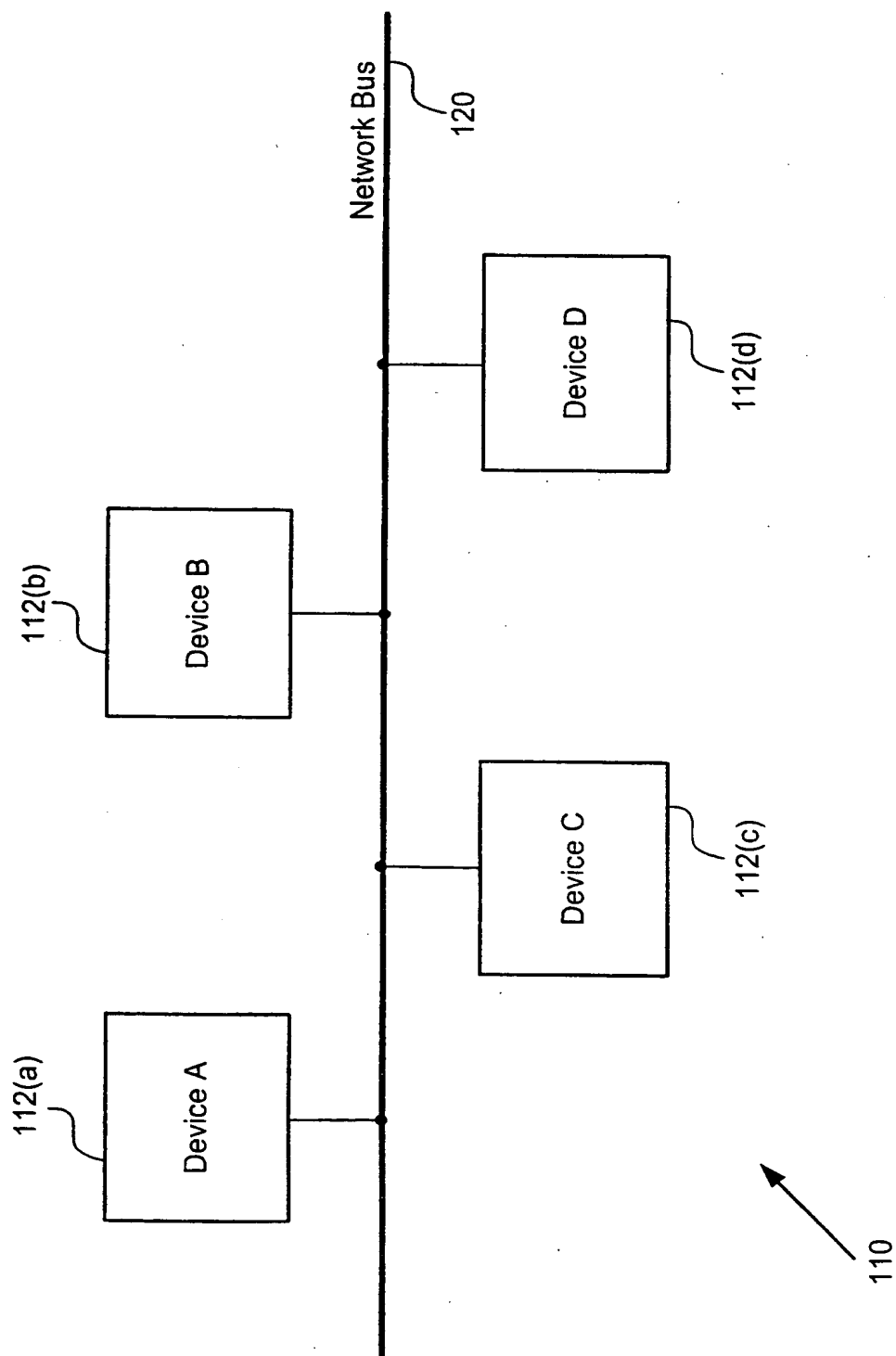


FIG. 1

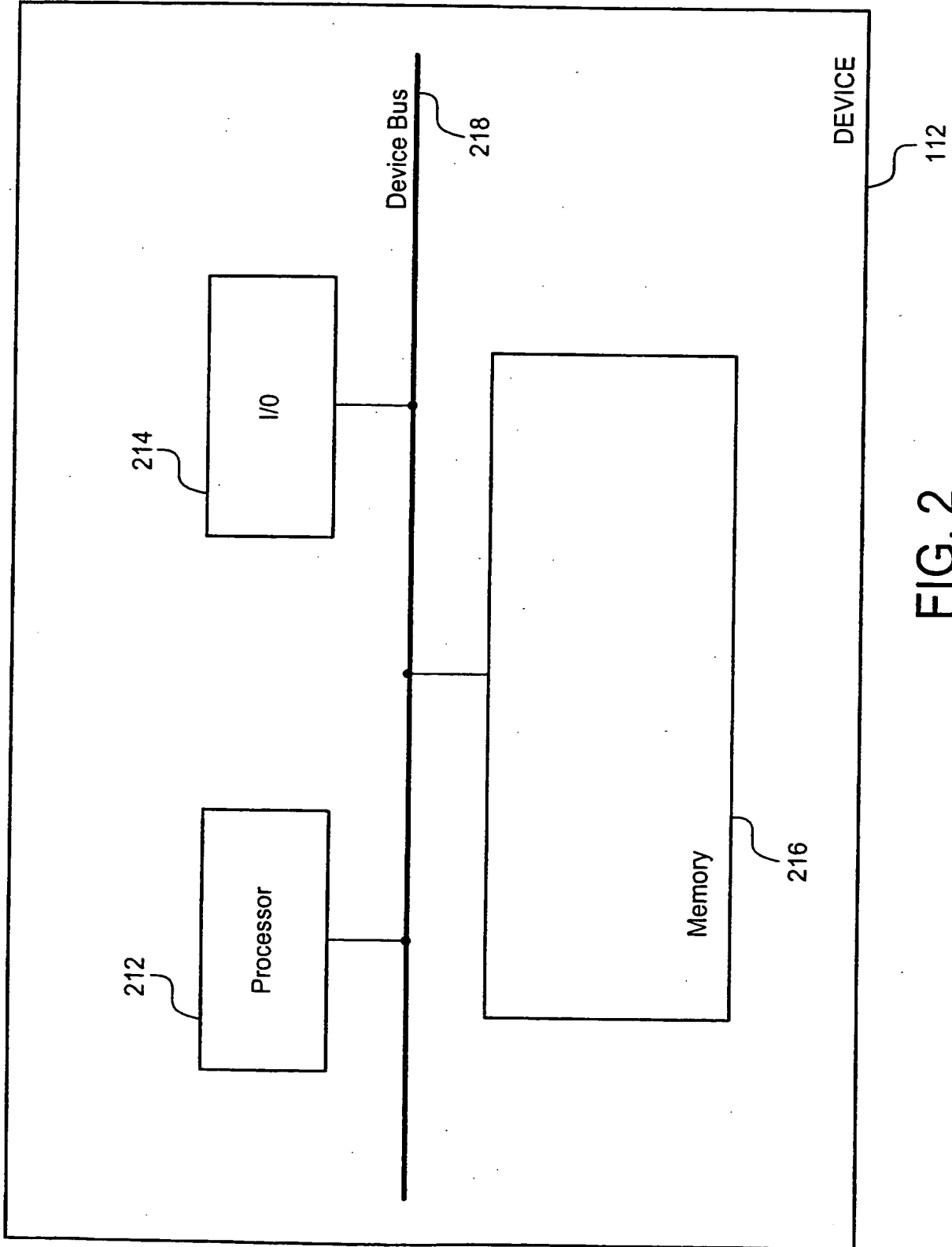


FIG. 2

3/10

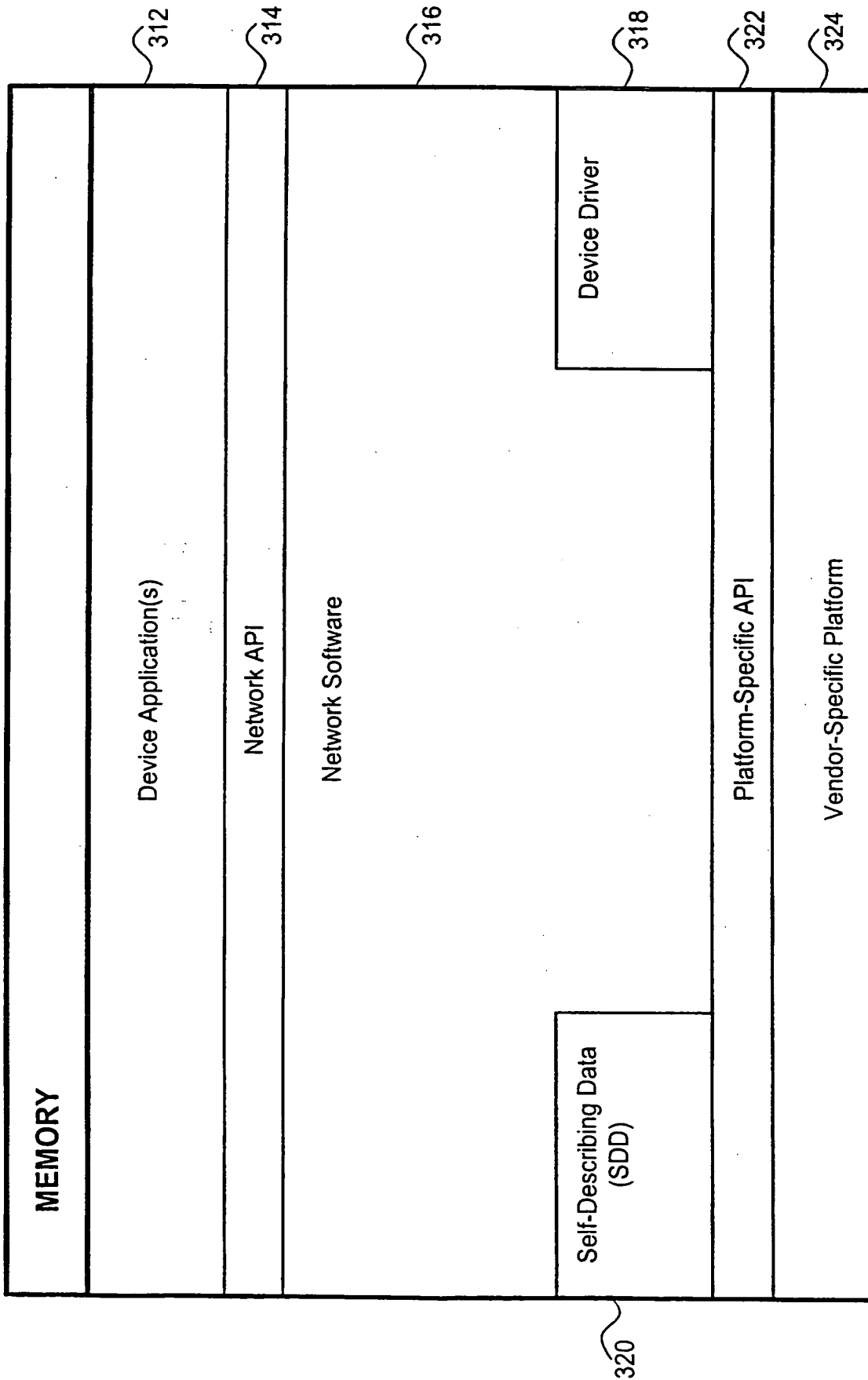


FIG. 3

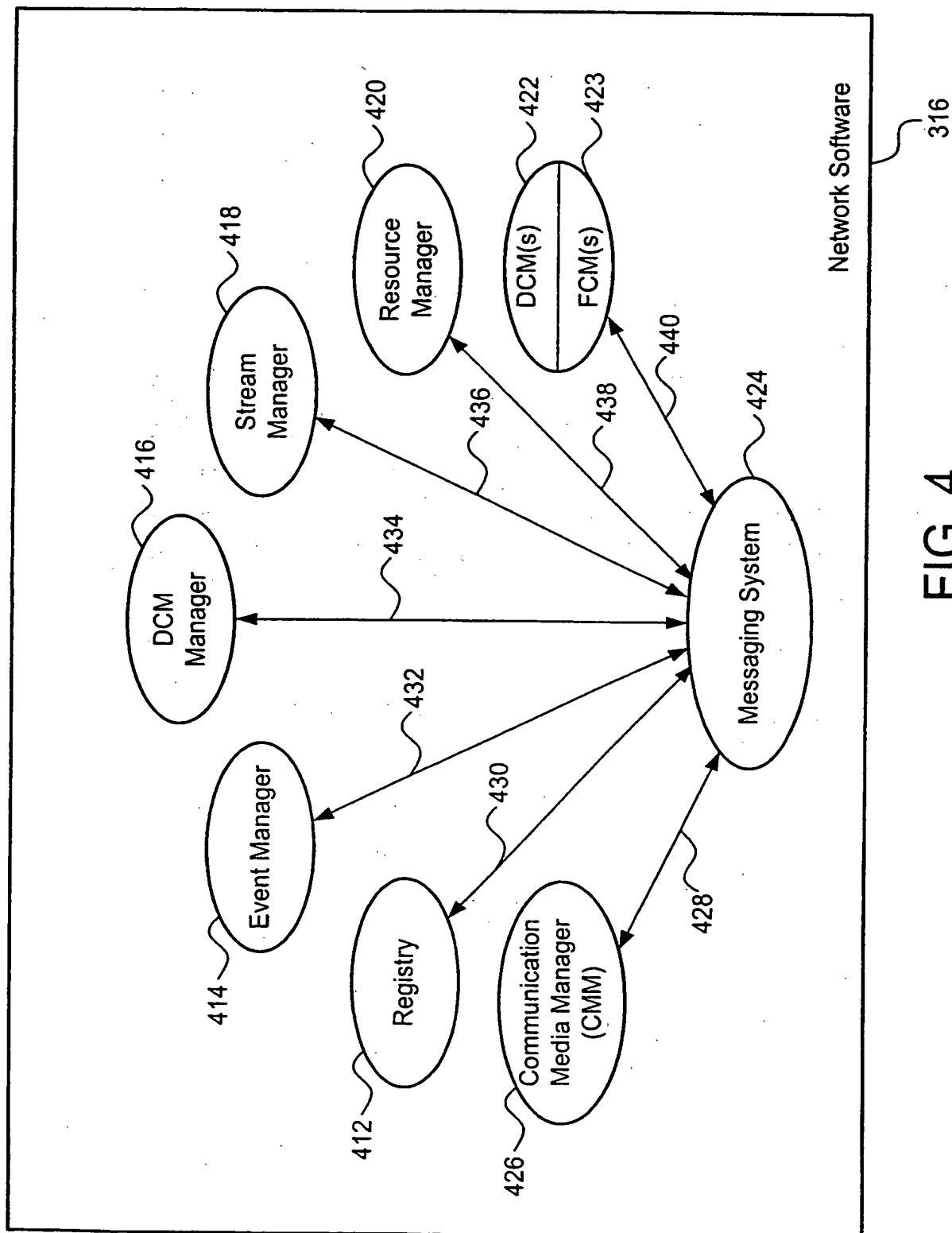


FIG. 4

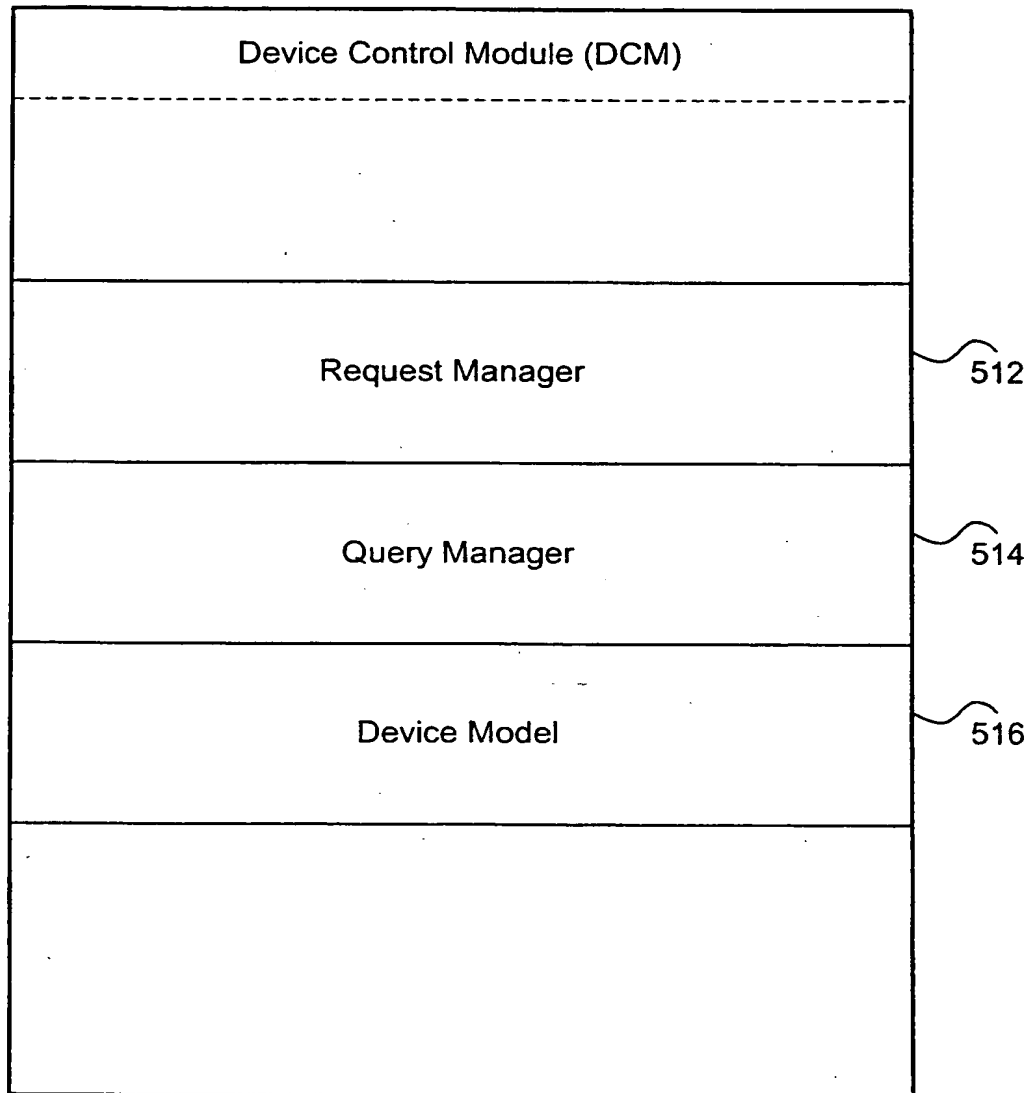
422

FIG. 5

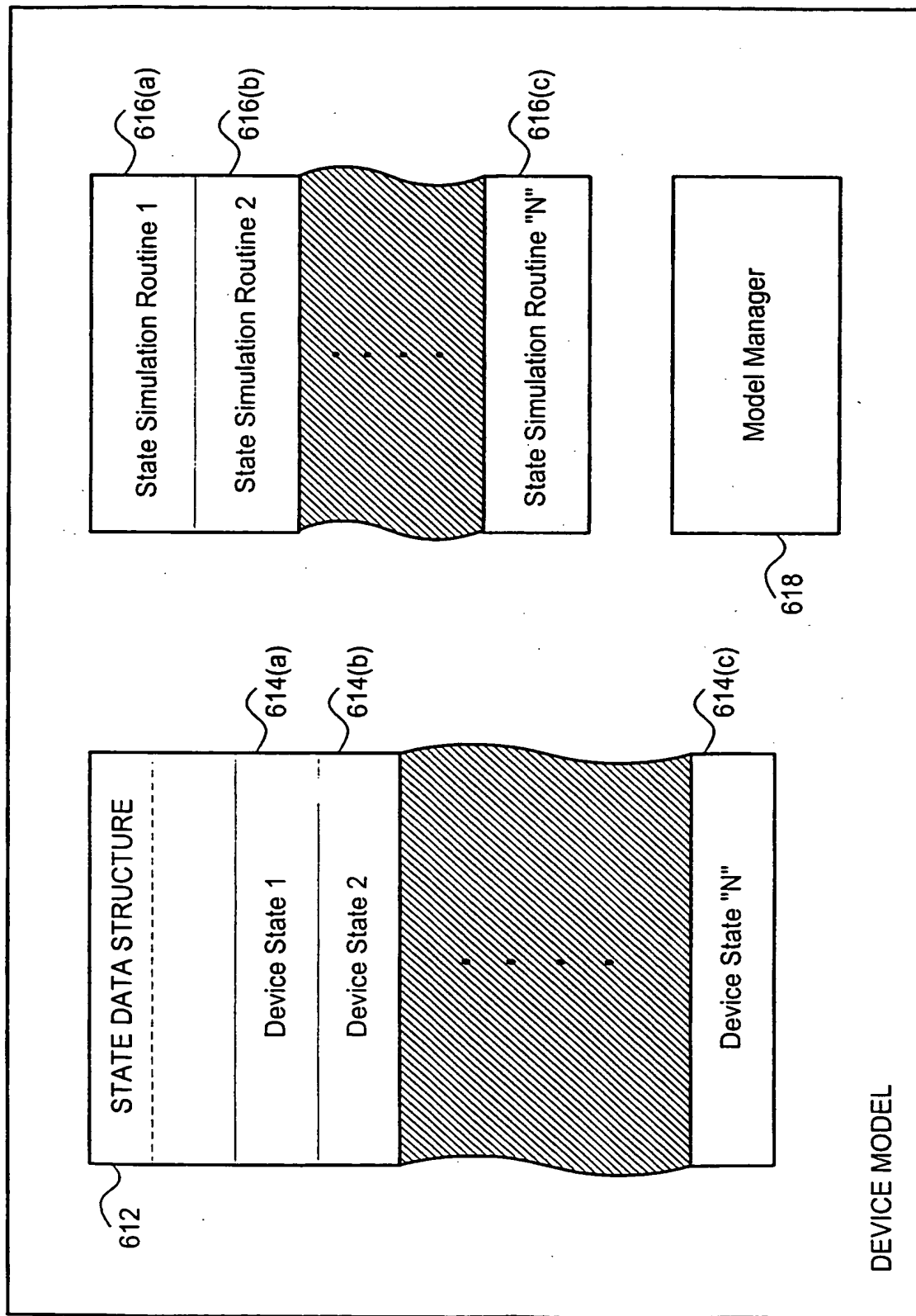


FIG. 6

7/10

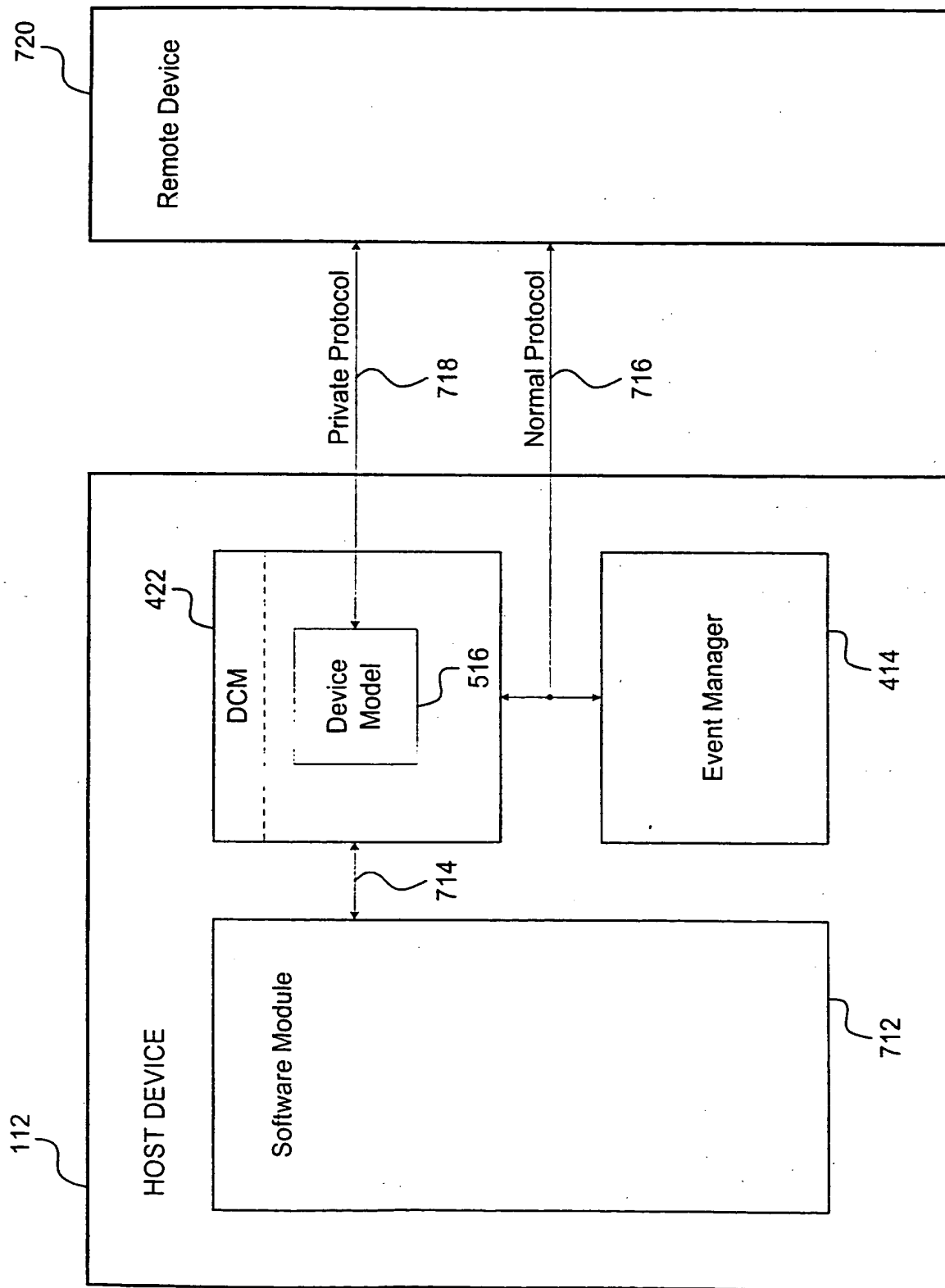


FIG. 7

8/10

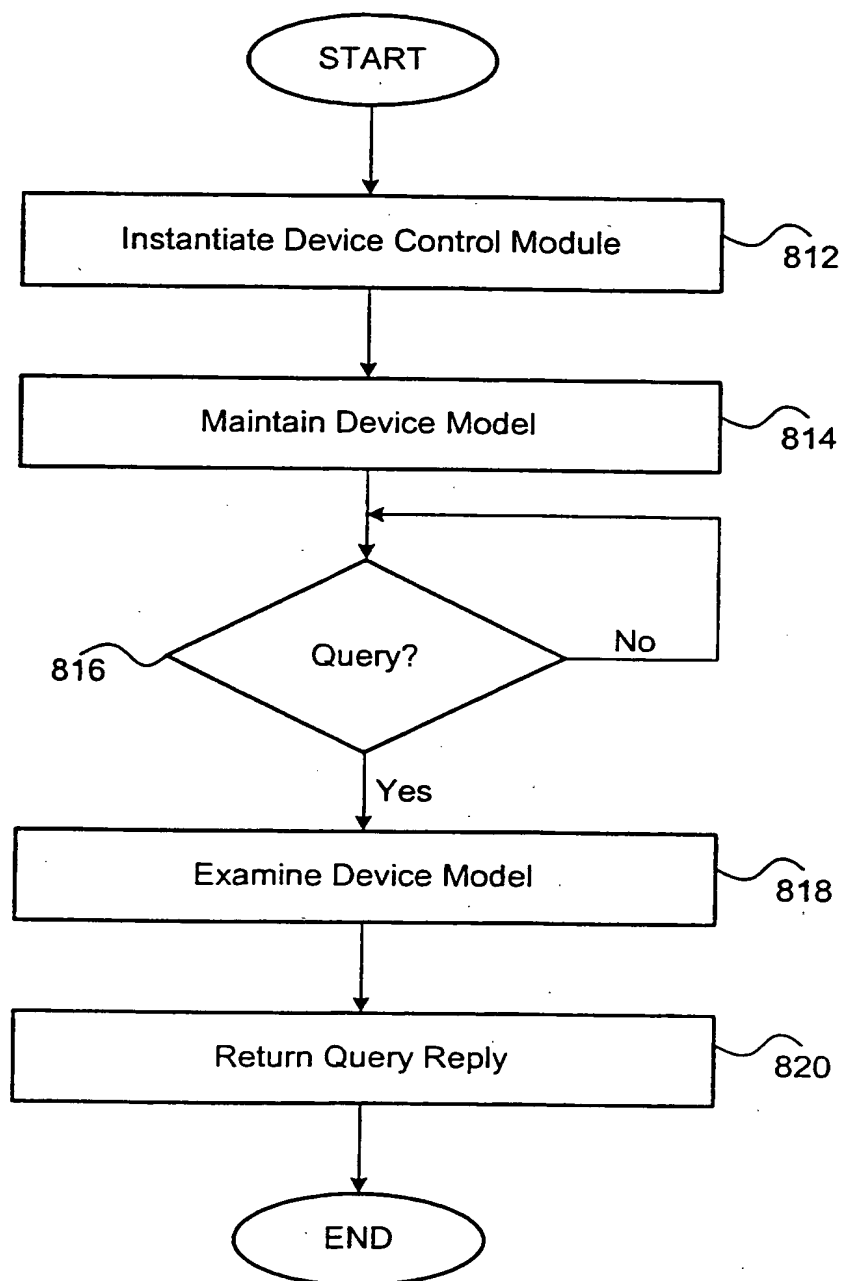


FIG. 8

9/10

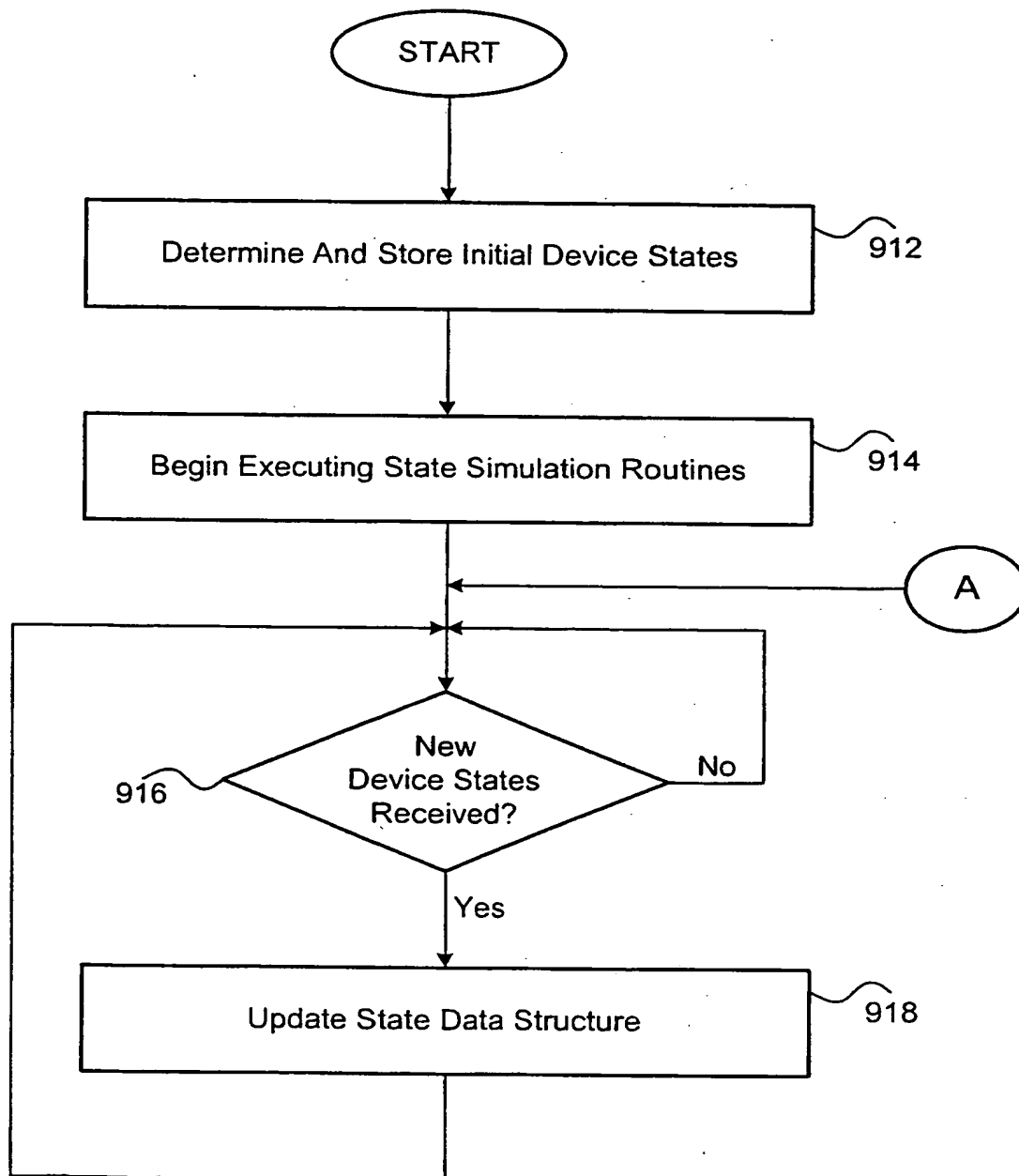


FIG. 9

10/10

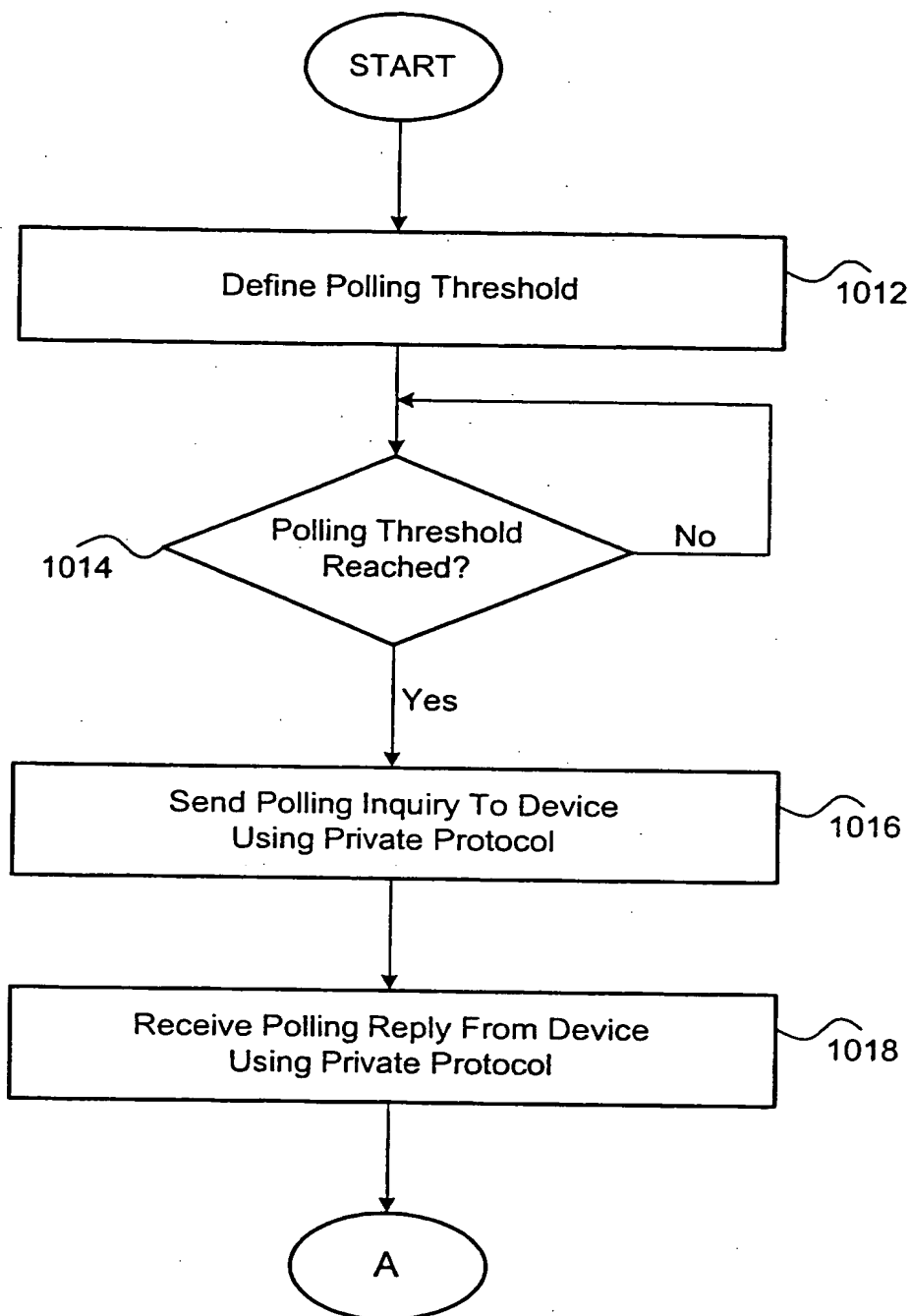


FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US00/21037

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 15/173

US CL : 709/223, 224

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709 223, 224

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,862,404 A (ONAGA) 19 January 1999, cols. 2-6.	1-46
A	US 6,003,064 A (WICKI et al) 14 December 1999.	1-46
A	US 5,592,620 A (CHEN et al) 07 January 1997.	1-46
A	US 5,764,913 A (JANCKE et al) 09 June 1998.	1-46



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y* document of particular relevance, the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*G* document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

19 SEPTEMBER 2000

Date of mailing of the international search report

20 NOV 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

KENNETH FIELDS

Telephone No. (703) 305-3900